



TEC

Tecnológico
de Costa Rica

TECNOLÓGICO DE COSTA RICA

Ingeniería en Computación

Bases de datos I / Grupo 1

Investigación: Bases de Datos NoSQL: MongoDB.

Gabriel Ramírez Ramírez – 201020244.

Junior Herrera – 2014110993

José Arturo Luna –2014110993

Fecha de entrega: 29 de noviembre de 2017.

II Semestre, 2017.

Profesor: William Mata Rodríguez.

Contenido

Introducción.....	3
Enunciado.....	4
Desarrollo	6
Diseño Conceptual.....	9
Diseño Lógico.	10
Estructura de los Archivos: JSON.....	11
Conclusiones.....	12
RUBRICAS DE EVALUACIÓN.....	13

Introducción

El siguiente documento busca detallar de la mejor forma posible como se realizó el proyecto de investigación en MongoDB.

En este trabajo se desarrollaron conocimientos fundamentales que se puedan tener sobre las denominadas Bases de Datos NoSQL. Y la importancia que han adquirido en los últimos años, gracias a la mayor capacidad de almacenamiento y el bajo costo que este tiene, así como velocidad a la hora de generar un reporte (consulta) sobre esos datos.

Además, se necesitó adquirir conocimientos del DBMS para MongoDB, así como herramientas visualmente muy útiles para que permitan una mejor administración y ejecución de la Base de Datos.

Se inició el proceso de elaboración de la solución de software haciendo el levantamiento de los requerimientos establecidos por el profesor y elaborando un modelo conceptual inicial con la estructura que van a tener los diferentes datos de la aplicación. Luego se procedió a la implementación de la solución en Mongo, así como la creación de una aplicación creada en Java.

Git del proyecto: <https://github.com/GabrielRA/ProyectoBD-MONGO/>

Enunciado

El resumen va a tener por cada partido:

- + número del partido: este número debe estar registrado en el sistema CAMPEONATO MUNDIAL (proyecto 2). Sino está registrado se retorna el mensaje “NO PUEDE REGISTRAR EL RESUMEN PORQUE EL PARTIDO AÚN NO ESTÁ EN EL SISTEMA”. En caso de que el partido esté registrado se despliegan los nombres de los equipos.
- + texto del resumen (tamaño variable, n caracteres).
- + videos de cada gol del partido o de partes importantes del partido (al menos 2 videos de 30 segundos mínimo cada uno).

Luego de registrar el resumen, los aficionados pueden hacer comentarios. Cada comentario va a tener:

- + número de comentario para este partido: es un entero secuencial (o correlativo) a partir de 1. La aplicación debe asignar este número único para cada comentario.
- + aficionado que hizo el comentario. Opcionalmente su foto y su correo electrónico.
- + fecha (día, mes, año) y hora (minutos, segundos) del comentario.
- + texto del comentario (tamaño variable, n caracteres). Una vez registrado un comentario no se puede cambiar ni borrar.
- + los comentarios pueden ser independientes entre ellos o seguir un hilo, es decir, pueden estar contestando (reply) a otro comentario.

Los aficionados están en una lista que va a tener:

- + código único del aficionado (máximo 15 caracteres)
- + contraseña: se guarda físicamente con un algún método de encriptación.
- + foto del aficionado. Se despliega en los comentarios donde aparezca el aficionado si así lo indica el registro de aficionados.
- + indicador de despliegue de foto.
- + correo electrónico del aficionado. Se despliega en los comentarios donde aparezca el aficionado si así lo indica el registro de aficionados.
- + indicador de despliegue de correo.

POR HACER:

- Diseño de la base de datos:
 - o Modelado de datos en MongoDB.

- Esquema básico de seguridad que permita ingresar a las funcionalidades de la aplicación.

- o CRUD de la lista de aficionados (lista de usuarios). Los aficionados se autoregistran. Pueden hacer modificaciones. El borrado no es físico, es lógico y para ello se guarda la fecha y hora del borrado. Una vez borrado el código del aficionado no puede volver a usarse. En los comentarios donde aparezca un código borrado se pondrá el texto BORRADO junto con la fecha y hora del mismo.

- CRUD del resumen: solo el aficionado con el código ADMINISTRADOR lo puede hacer. En caso de que existan comentarios se deben desplegar en orden descendente por fecha y hora, es decir, los más recientes de primero.
- CRUD de los comentarios hechos por los aficionados. Para hacer un comentario primeramente se pide el número del partido del cual se despliega la siguiente información en este orden:
 - o Nombres de los equipos
 - o Texto del resumen del partido
 - o Videos
 - o Espacio para el comentario que puede hacer el aficionado

Desarrollo

Para poder desarrollar el Proyecto lo primero que se hizo fue buscar información y documentación sobre el diseño de la Bases de Datos en MongoDB y cuál era el lenguaje de consulta de la Base de Datos.

Temas Investigados

MongoDB: Es un motor de Base de Datos que permite desarrollar sistemas NoSQL. Es una base de datos orientada a documentos. Esto quiere decir que en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON. Y no es necesario seguir un esquema.

Mongod: Es un servicio que empezará a escuchar peticiones por el puerto 27017. Es importante indicar el parámetro de la ruta dónde se almacenarán los ficheros de nuestra base de datos.

mongo: Este es el servicio que permite levantar el DBMS ya que permite el acceso y trabajar directamente en la Base de Datos.

RoboMongo: Es una GUI que permite controlar y acceder de manera más a la Base de Datos. RoboMongo crea una conexión con Mongo DB y poder así acceder a los distintos datos.

Importar Datos: MongoDB proporciona una aplicación útil para importación de los datos hacia la Base de Datos.

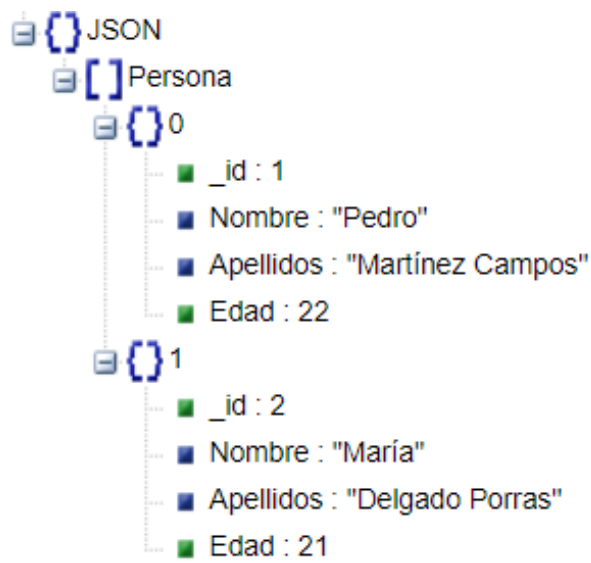
NetBeans: Este software permite la elaboración y ejecución de código en el lenguaje JAVA, así como herramientas para la creación de la GUI de una manera sencilla y fácil de usar. También brinda las herramientas necesarias para manipular los datos de la Base de Datos en MongoDB.

mongo-java-driver-2.9.3.jar: Es una librería que existe en JAVA para poder controlar la conexión a la Base de Datos, así como la manipulación de datos: texto, números, imágenes y video.

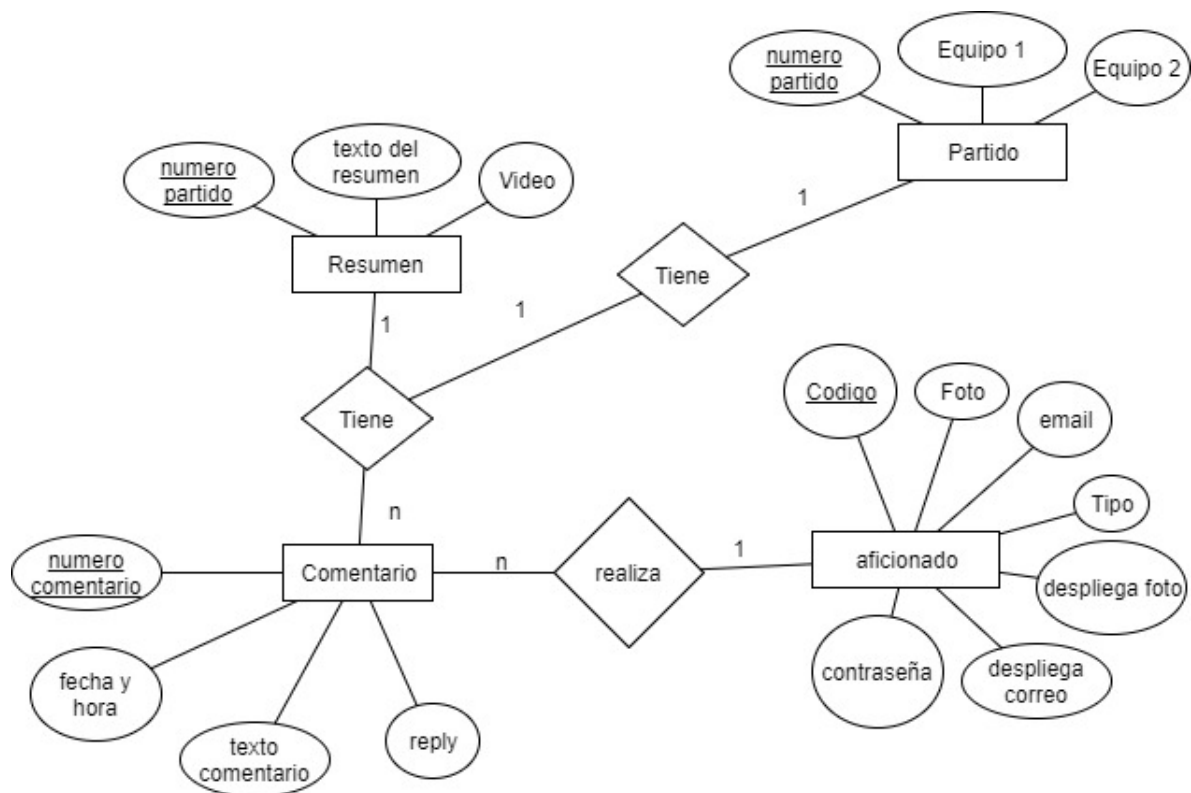
Git: Se investigó sobre Git como herramienta para controlar de versiones y como el repositorio en el cual todos subíamos los cambios realizados en el proyecto, debido a que es una muy buena herramienta para poder implementar trabajo en equipo.

Software de manejo de Versiones: Para el desarrollo del proyecto decidimos usar la plataforma de GitHub además de que el lenguaje de manipulación de la base de datos usada fue NetBeans.

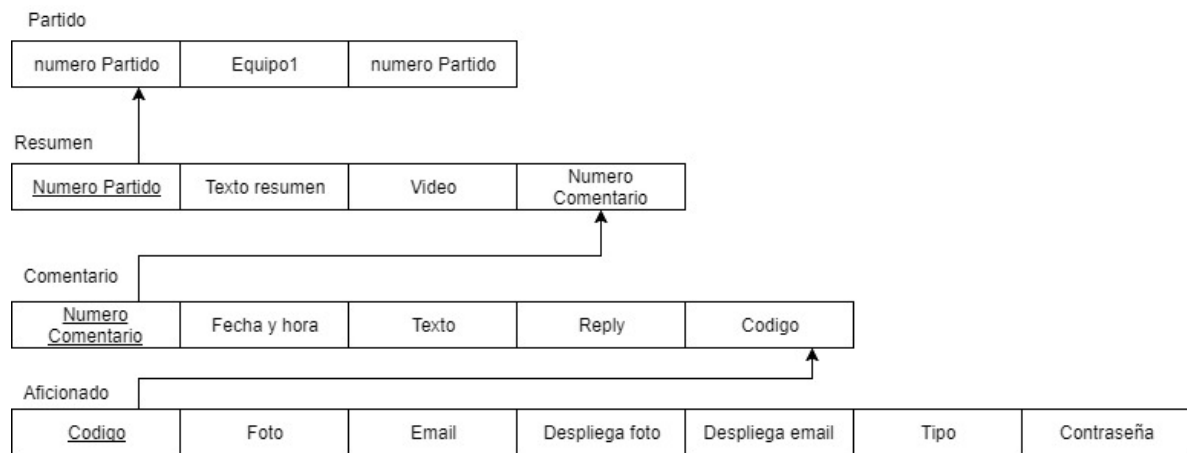
Json: Es un tipo de archivo que permite almacenar colecciones de datos. El formato Json nace del lenguaje de programación Javascript y se usa para el intercambio de los datos.



Diseño Conceptual.



Diseño Lógico.



Estructura de los Archivos: JSON.

Aficionads.json

```
{ "idPartido" : 2, "equipo1" : "Brasil", "equipo2" : "Croacia" }
{ "idPartido" : 2, "equipo1" : "Mexico", "equipo2" : "Camerún" }
{ "idPartido" : 3, "equipo1" : "España", "equipo2" : "Holanda" }
{ "idPartido" : 4, "equipo1" : "Chile", "equipo2" : "Australia" }
{ "idPartido" : 5, "equipo1" : "Colombia", "equipo2" : "Grecia" }
{ "idPartido" : 6, "equipo1" : "Uruguay", "equipo2" : "Costa Rica" }
{ "idPartido" : 7, "equipo1" : "Inglaterra", "equipo2" : "Italia" }
{ "idPartido" : 8, "equipo1" : "Costa de Marfil", "equipo2" : "Japón" }
{ "idPartido" : 9, "equipo1" : "Suiza", "equipo2" : "Ecuador" }
{ "idPartido" : 10, "equipo1" : "Francia", "equipo2" : "Honduras" }
{ "idPartido" : 11, "equipo1" : "Argentina", "equipo2" : "Bosnia" }
{ "idPartido" : 12, "equipo1" : "Alemania", "equipo2" : "Portugal" }
{ "idPartido" : 13, "equipo1" : "Irán", "equipo2" : "Nigeria" }
{ "idPartido" : 14, "equipo1" : "Ghana", "equipo2" : "Estados Unidos" }
{ "idPartido" : 15, "equipo1" : "Mexico", "equipo2" : "Camerún" }
...
```

Conclusiones

Aprendimos que al utilizar Github y se facilita el trabajo debido a que varias personas podían avanzar diferentes partes para luego solo subir los cambios, además de que se podía manejar y administrar mejor el proyecto que se estaba desarrollando.

El controlador de versiones permite llevar un control sobre la participación de los distintos miembros del equipo de desarrollo del proyecto.

Se tuvieron problemas con las operaciones Pull y Push de Github, ya que hubieron varias ocasiones que duraba mucho tiempo en cargar el repositorio, pero lo más difícil fue que nos funcionara a todos los integrantes del Grupo.

Otro error común con el controlador de versiones eran que estás no coincidían entonces había que hacer merge y corregir a una versión anterior.

En java la conexión con la base de datos al principio fue un problema, pero luego de investigar un poco se pudo resolver fácilmente, además java al ser un lenguaje muy utilizado se encontró con facilidad la información que buscábamos.

MongoDB posee un lenguaje de Consulta muy distinto al conocido de SQL. Es como estar escribiendo código en Javascript.

Para la importación de datos de un archivo JSON a la Base de Datos es necesario ubicar dicho archivo en la misma dirección donde se encuentra el mongoimport.

Mongo no acepta documentos de más de 16 MB por lo que para almacenar imágenes y videos fue necesaria la utilización del paquete GridFS. El cual divide el tamaño de un archivo y lo hace reubicable dentro de la Base de Datos.

RUBRICAS DE EVALUACIÓN.

Concepto	Puntos	Puntos Obtenidos	Avance 100/%/0	Análisis de Resultados
Diseño de la Base de Datos	5		100	
CRUD Aficionados	10		100	
CRUD Resumen	35		80	
CRUD Comentario	25		80	
Validaciones de datos y procesos	10		100	
OPERACIONES MONGO: consultas en vivo que van a desarrollar los estudiantes de forma individual según el modelo implementado por el grupo.	10*			
Ayuda	5		100	
TOTAL	100			
Partes desarrolladas adicionalmente			100	